

-
-
-
-
-

IF-705 – Automação Inteligente

Algoritmos Genéticos

Aluizio Fausto Ribeiro Araújo
Universidade Federal de Pernambuco
Centro de Informática - CIn
Departamento de Sistemas da Computação
aluizioa@cin.ufpe.br



-
-
-
-
-

Conteúdo

- Introdução
- Exemplo introdutório.
- Possibilidades de representações
- Outro exemplo

Introdução

- Algoritmo Evolucionário (AE) mais conhecido e mais usado.
- Desenvolvido nos EUA nos anos 1970.
- Pioneiros: John Holland, Kenneth A. DeJong, David E. Goldberg.
- Características dos modelos iniciais:
 - Lento;
 - Maior importância para o cruzamento que para a mutação;
 - Aplicado à problemas combinatórios;
 - Aparecimento de variações dos operadores.

Introdução

- Holland propôs um dos primeiros modelos foi o Algoritmo Genético Simples (SAG), caracterizado por:
 - Representação binária;
 - Seleção de pais proporcional à aptidão.
 - Baixa probabilidade de mutação.
 - Recombinação geneticamente inspirada.
 - Seleção generacional de sobreviventes.
- Outros modelos de AGs podem variar em:
 - Representações;
 - Operadores de variação;
 - Mecanismos de seleção.

Algoritmo Genético Simples (SGA)

Tabela com Sumário Técnico

Representação	Strings binários
Recombinação	N-pontos ou uniforme
Mutação	Troca de bits com probabilidade fixa
Seleção de pais	Proporcional à aptidão
Seleção de sobreviventes	A prole substitui a geração anterior
Especialidade	Ênfase no cruzamento

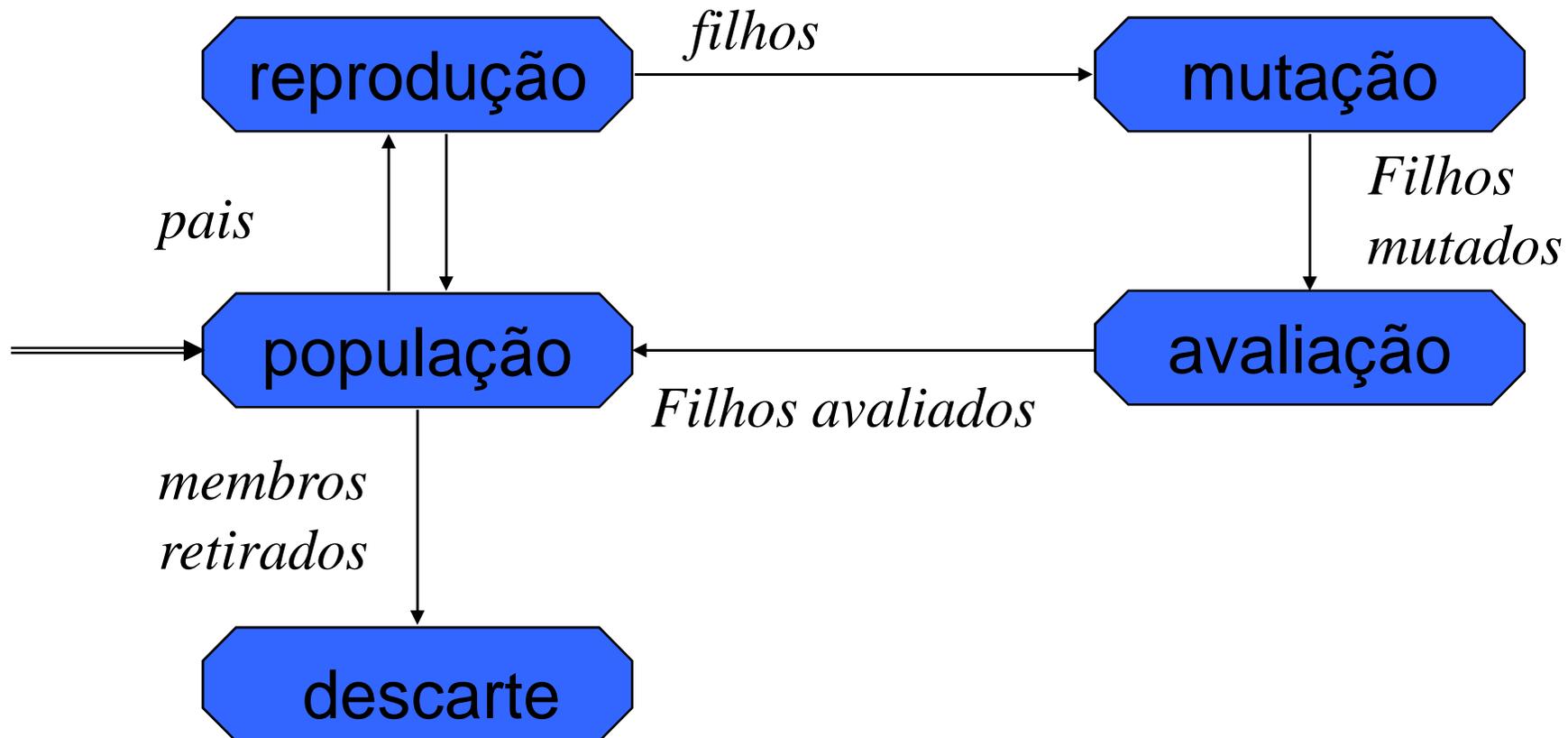
Algoritmo Genético Simples (SGA)

Ciclo de Evolução

1. Selecione pais para conjunto de acasalamento:
 1. Tamanho do conjunto de acasalamento = tamanho da população.
2. Misture o conjunto de acasalamento.
3. Para cada par consecutivo, aplique cruzamento com probabilidade p_c , se não cruzar, copie os pais.
4. Para cada filho, aplique o operador de mutação (troca de bit com probabilidade p_m independente para cada bit).
5. Substitua toda a população pela prole resultante.

Algoritmo Genético Simples (SGA)

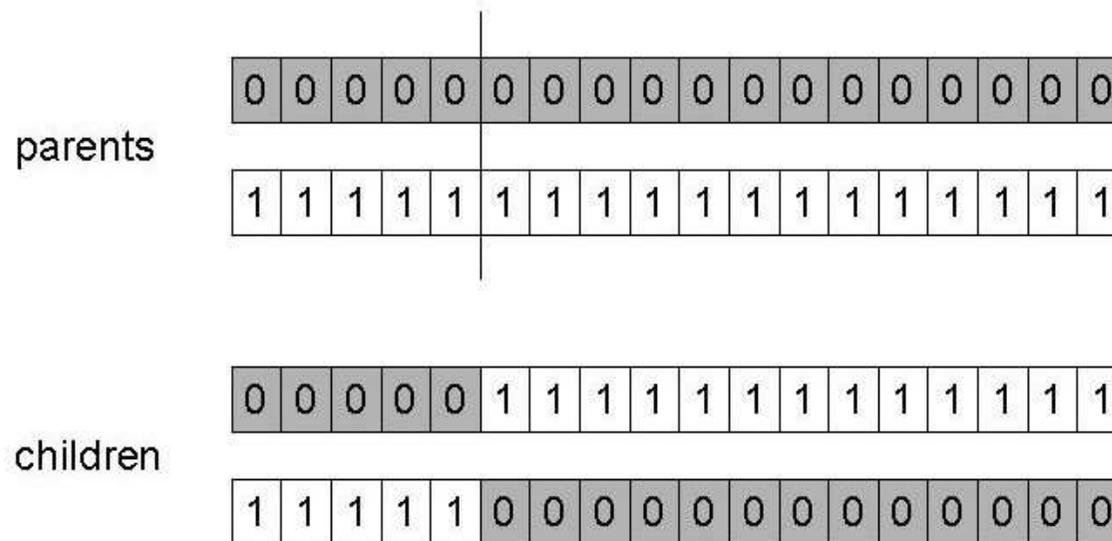
Ciclo de Evolução



Algoritmo Genético Simples (SGA)

Cruzamento de um Ponto

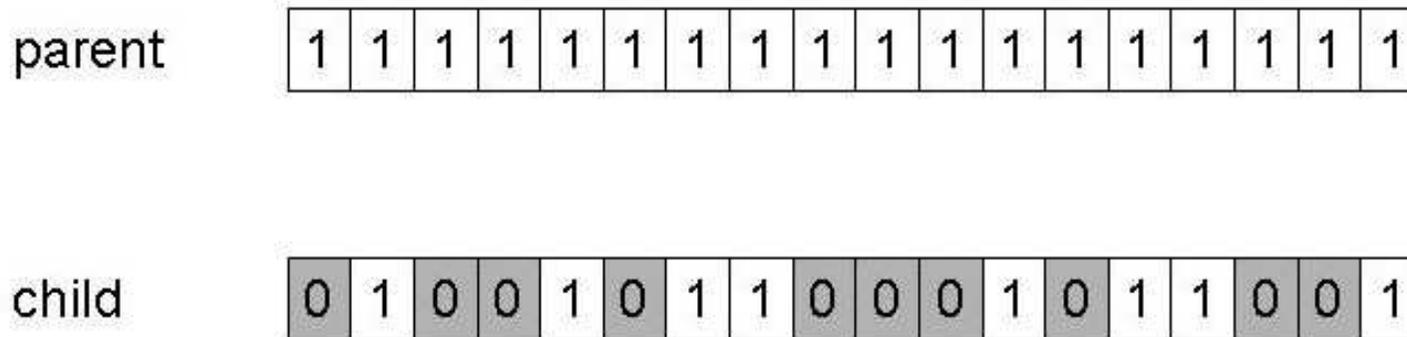
- Escolha um ponto aleatório em um par de pais.
- Divida o genótipo dos pais neste ponto.
- Crie filhos pela troca das caldas dos pais.
- A probabilidade de cruzamento, p_c tem intervalo típico (0.6, 0.9).



Algoritmo Genético Simples (SGA)

Mutação

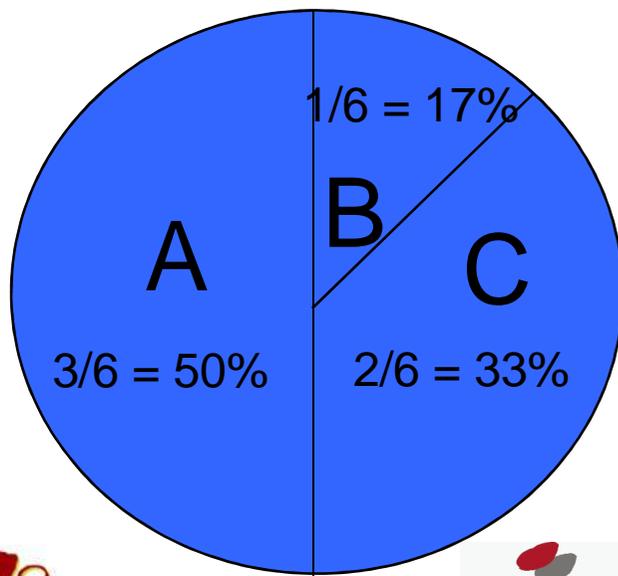
- Altere cada gene independentemente, com probabilidade P_m .
- p_m é a taxa de mutação
 - Tipicamente ela é definida entre $1/(\text{tamanho da população})$ e $1/(\text{comprimento do cromossomo})$.



Algoritmo Genético Simples (SGA)

Seleção

- Premissa: Indivíduos mais aptos têm mais chance de serem selecionados (proporcionais à aptidão):
 - Atribua para cada indivíduo uma parte da roleta.
 - Gire-a n vezes para selecionar n indivíduos.



←
aptidão(A) = 3
aptidão(B) = 1
aptidão(C) = 2

Exemplo: O Problema MAXONE

- Maximizar o número de bits com valor um em um string de bits de l dígitos binários.
 - Problema muito simples cuja resposta é conhecida;
 - Serve para validar o funcionamento do modelo;
 - Pode ser entendido como maximizar o número de respostas corretas, codificadas por 1, para questões pré-determinadas.

Exemplo: O Problema MAXONE

- Cada indivíduo é codificado como um string de l dígitos binários;
- A função de aptidão f de um candidato a solução é a soma dos números um que estão em seu genótipo;
- Parâmetros constantes:
 - Tamanho da população: $n = 6$;
 - Número de genes por indivíduo: $l = 10$.

Exemplo: O Problema MAXONE

Inicialização

- Lançamento da moeda por 60 vezes para ter a população inicial:

$$s_1 = 1111010101 \quad f(s_1) = 7$$

$$s_2 = 0111000101 \quad f(s_2) = 5$$

$$s_3 = 1110110101 \quad f(s_3) = 7$$

$$s_4 = 0100010011 \quad f(s_4) = 4$$

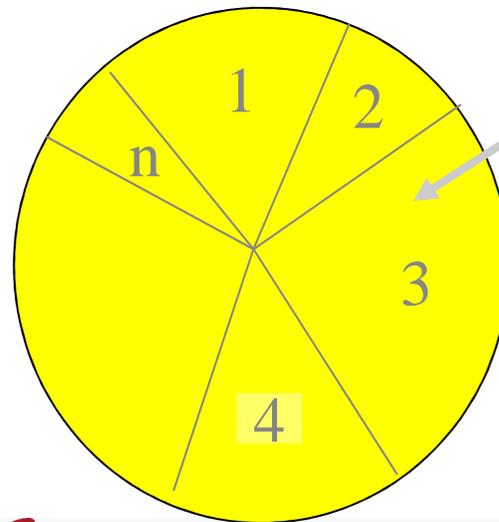
$$s_5 = 1110111101 \quad f(s_5) = 8$$

$$s_6 = 0100110000 \quad f(s_6) = 3$$

Exemplo: O Problema MAXONE

Seleção de Pais

- A seleção de pai é proporcional à aptidão, implementada pelo método da roleta:
 - Sorteia-se indivíduos para ser pais até ter o número de indivíduos da população (cada par de pais gera dois filhos).



Área é proporcional ao valor da aptidão

Exemplo: O Problema MAXONE

Seleção de Pais

- Conjunto de pais selecionados:

$$s_1^{\setminus} = 1111010101 \quad (s_1)$$

$$s_2^{\setminus} = 1110110101 \quad (s_3)$$

$$s_3^{\setminus} = 1110111101 \quad (s_5)$$

$$s_4^{\setminus} = 0111000101 \quad (s_2)$$

$$s_5^{\setminus} = 0100010011 \quad (s_4)$$

$$s_6^{\setminus} = 1110111101 \quad (s_5)$$

Exemplo: O Problema MAXONE Cruzamento

- Acasalamento dos strings para cruzamento:
 - Para cada casal considerado há a probabilidade ($p_c = 0,4$) associada dele realmente cruzar.
- De acordo com esta probabilidade, dois pares ($s_1^$, $s_2^$) e ($s_5^$, $s_6^$) serão realmente cruzados:
 - O primeiro par tem ponto de cruzamento igual a 2;
 - O segundo par tem ponto de cruzamento igual a 5;
 - Os outros possíveis pais geram cópias de si próprios.

Exemplo: O Problema MAXONE Cruzamento

Antes do cruzamento:

$$s_1^{\backslash} = 1111010101$$

$$s_2^{\backslash} = 1110110101$$

$$s_5^{\backslash} = 0100010011$$

$$s_6^{\backslash} = 1110111101$$

Após o cruzamento:

$$s_1^{\backslash\backslash} = 1110110101$$

$$s_2^{\backslash\backslash} = 1111010101$$

$$s_5^{\backslash\backslash} = 0100011101$$

$$s_6^{\backslash\backslash} = 1110110011$$

Exemplo: O Problema MAXONE

Mutação

- Cada bit de cada indivíduo da prole tem uma pequena probabilidade de haver mudança de bit: $p_m = 0,1$.

- Antes da mutação:

$$s_1^{''} = 1110110101$$

$$s_2^{''} = 1111010101$$

$$s_3^{''} = 1110111101$$

$$s_4^{''} = 0111000101$$

$$s_5^{''} = 0100011101$$

$$s_6^{''} = 1110110011$$

- Após a mutação:

$$s_1^{''' } = 1110100101 \quad f(s_1^{''' }) = 6$$

$$s_2^{''' } = 1111110100 \quad f(s_2^{''' }) = 7$$

$$s_3^{''' } = 1110101111 \quad f(s_3^{''' }) = 8$$

$$s_4^{''' } = 0111000101 \quad f(s_4^{''' }) = 5$$

$$s_5^{''' } = 0100011101 \quad f(s_5^{''' }) = 5$$

$$s_6^{''' } = 1110110001 \quad f(s_6^{''' }) = 6$$



Exemplo: O Problema MAXONE Término

- Em uma geração a aptidão completa da população cresceu de 34 para 37, 9% de seu valor.
- O processo será repetido até que um critério de parada seja encontrado.

Outras Possibilidades de Representação

- Várias representações:
 - String de Bits (0101 ... 1100)
 - Números inteiros (18 35 ... -41 13)
 - Números reais (43,2 -33,1 ... 0,0 89,2)
 - Permutações de elementos (E11 E3 E7 ... E1 E15)
 - Listas de regras (R1 R2 R3 ... R22 R23)
 - Partes de programas (árvores e suas partes)

Um Exemplo Simples

- Problema do caixeiro viajante (TSP):
- Determina um rota para um conjunto de cidades tal que:
 - Cada cidade seja visitada apenas uma vez;
 - A distância total da viagem seja minimizada.

Representação do TSP

- A representação é uma lista ordenada de números de cidades:

1) London 3) Dunedin 5) Beijing 7) Tokyo
2) Venice 4) Singapore 6) Phoenix 8) Victoria

Lista_de_cidades_1 (3 5 7 2 1 6 4 8)

Lista_de_cidades_2 (2 5 7 6 8 1 3 4)

Cruzamento no TSP

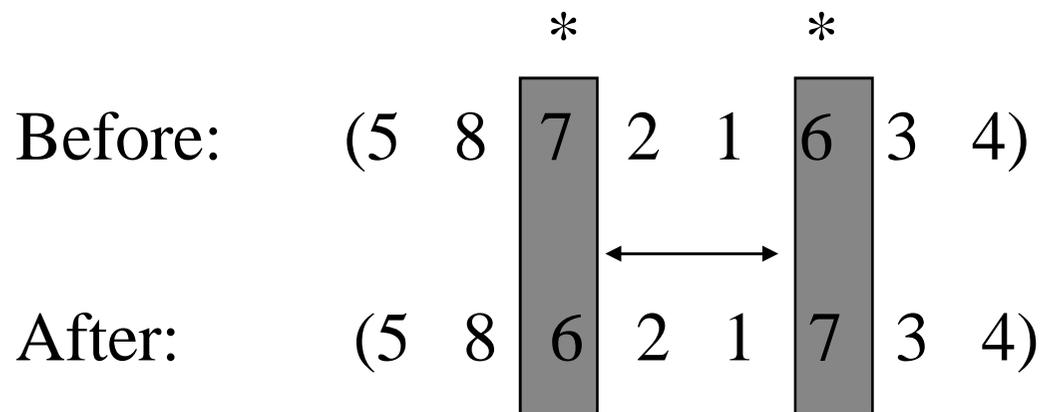
- O operador de cruzamento é resultado de operação de recombinação e complementação de genes sem duplicidade:

		*		*			
Pai1	(3	5	7	2	1	6	4 8)
Pai2	(2	5	7	6	8	1	3 4)
<hr/>							
filho	(5	8	7	2	1	6	3 4)

- Operador chamado cruzamento de Ordem 1.

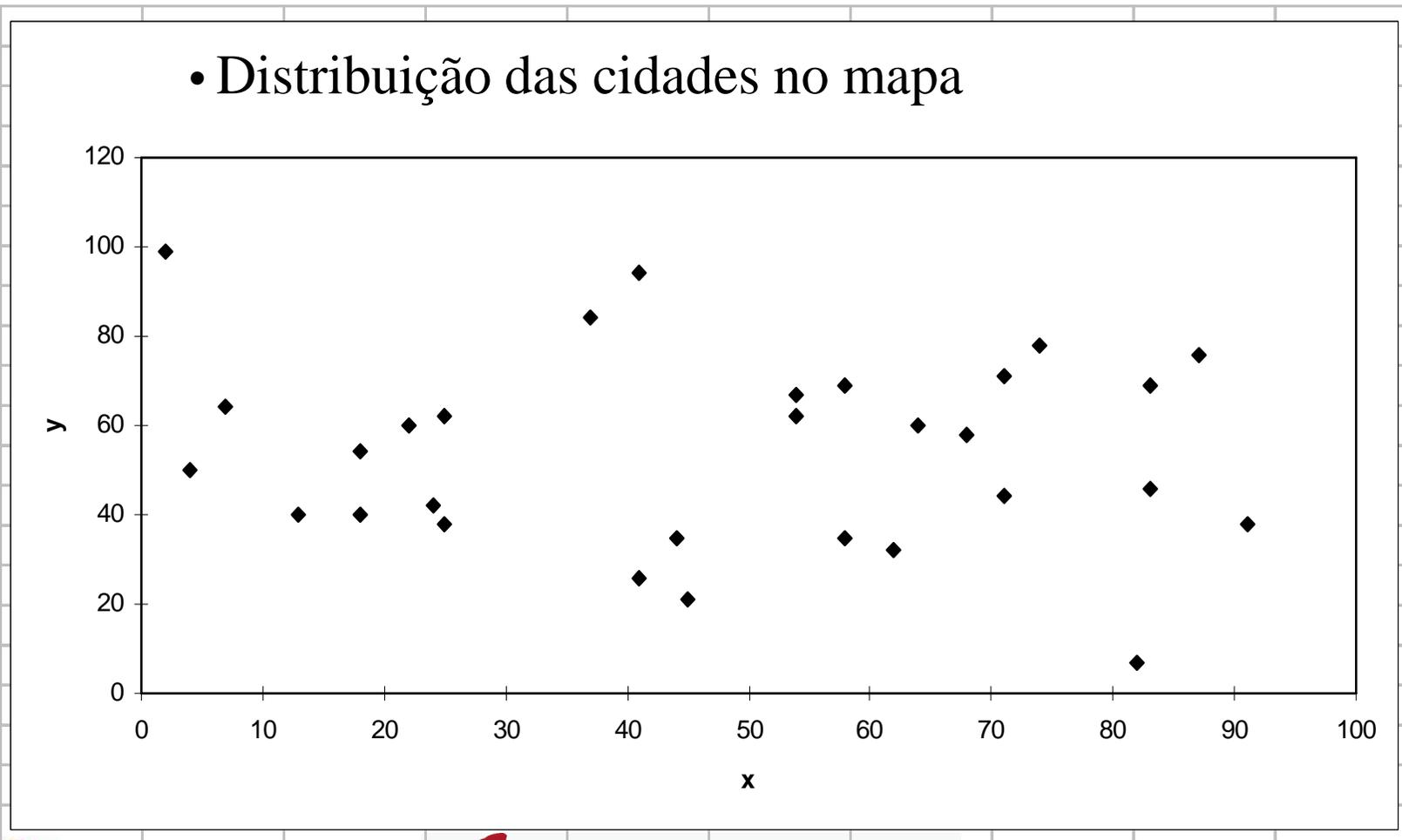
Mutação no TSP

- Compreende re-ordenamento da lista

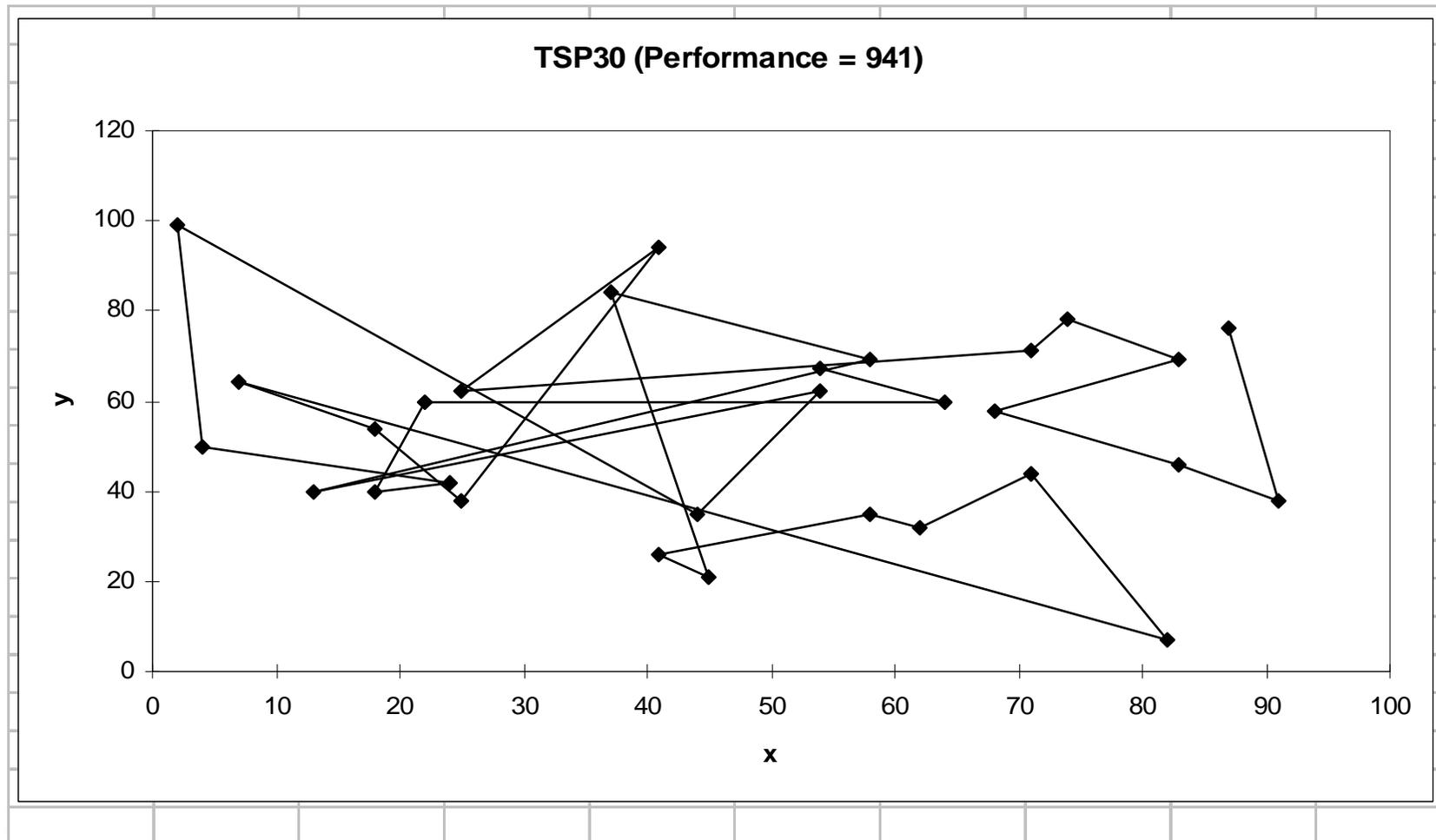


TSP para 30 Cidades

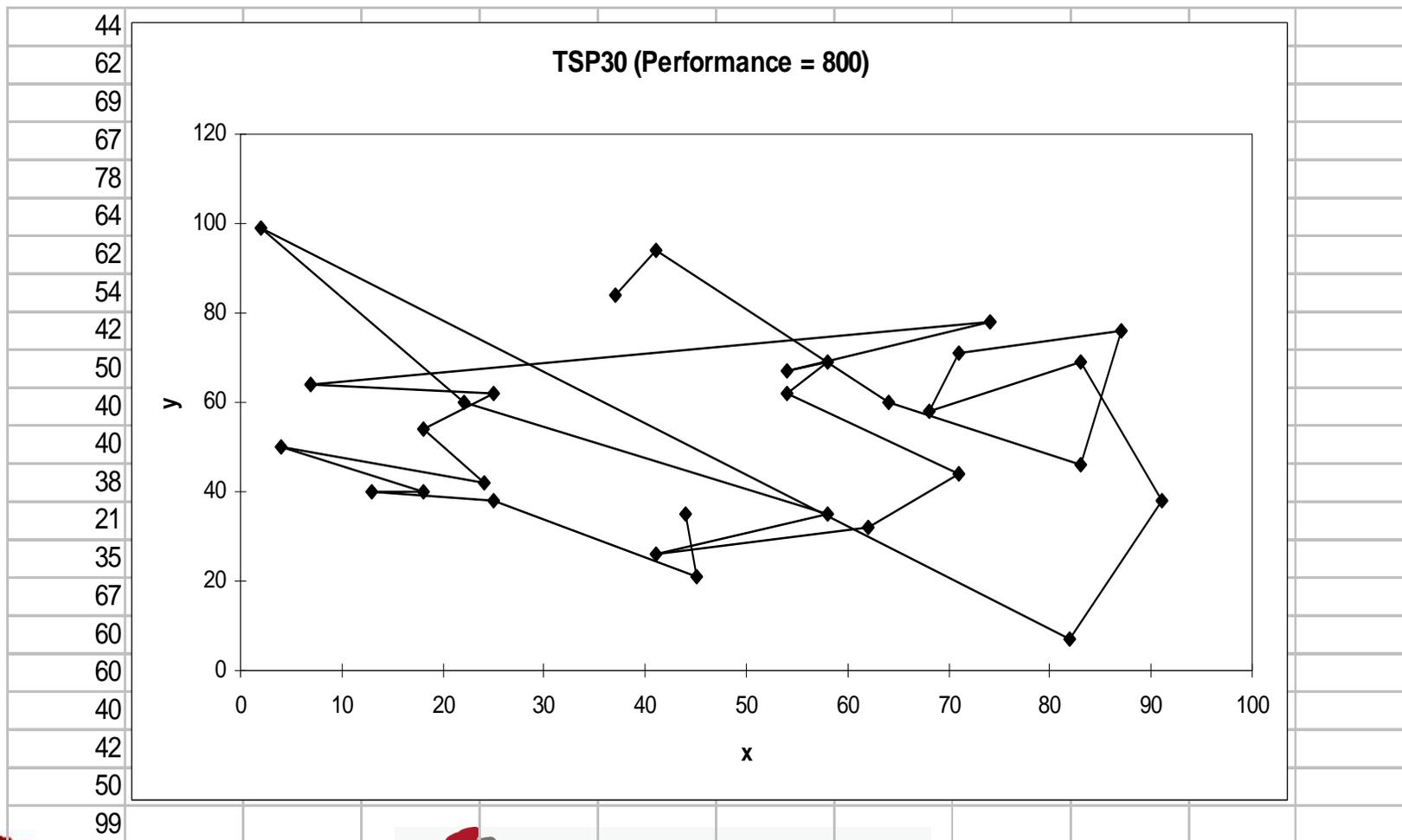
• Distribuição das cidades no mapa



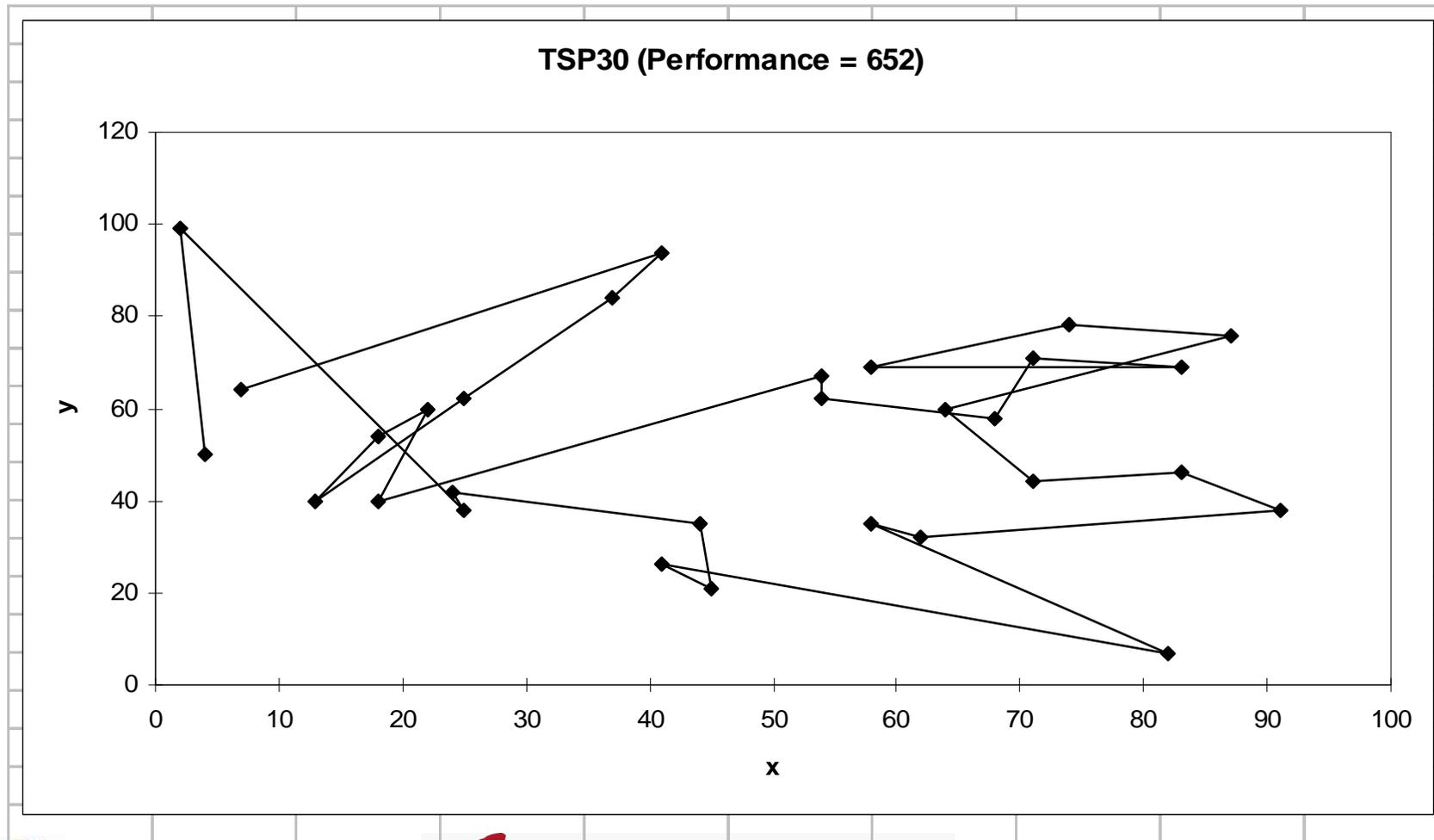
TSP: Primeira solução



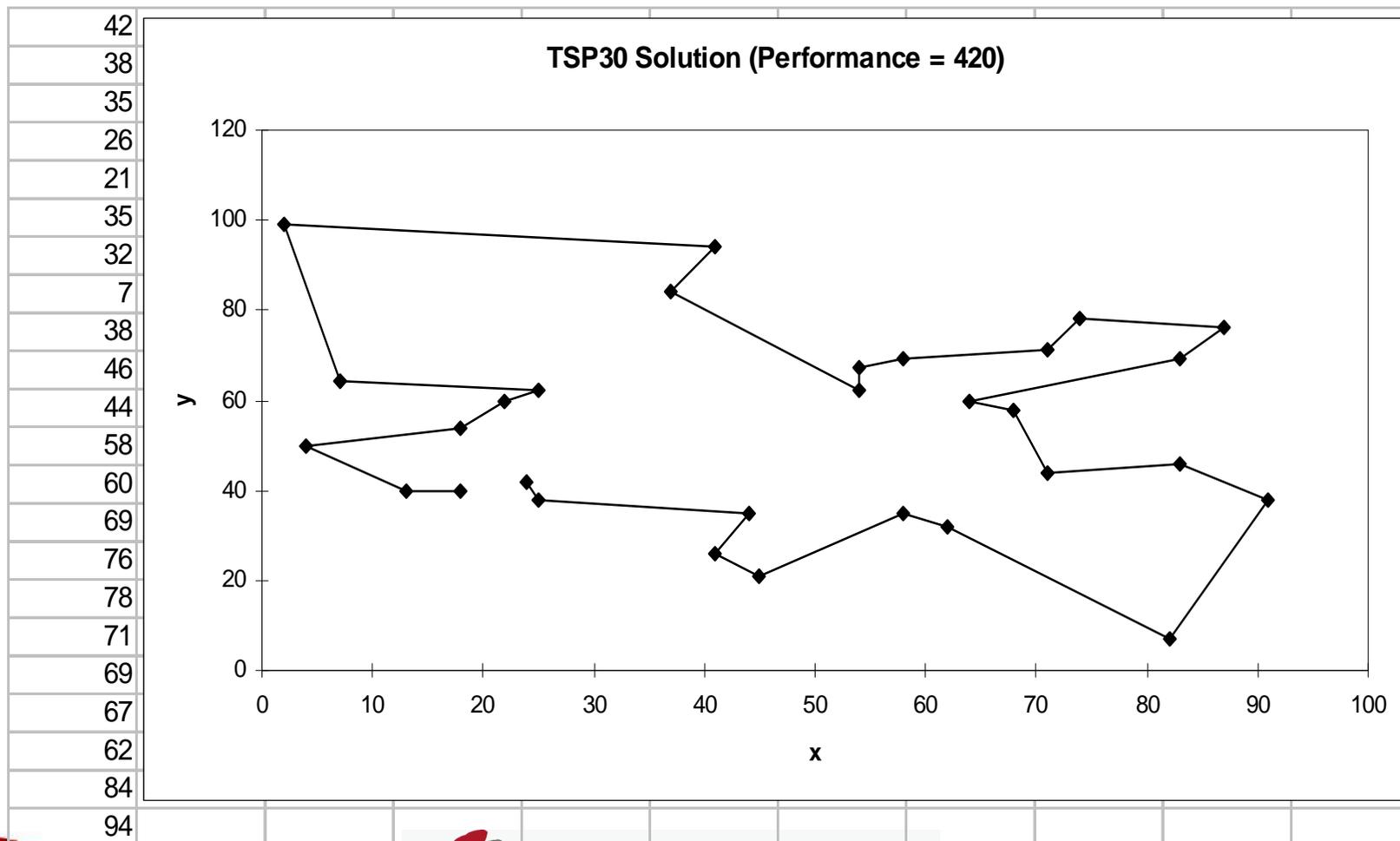
TSP: Segunda solução



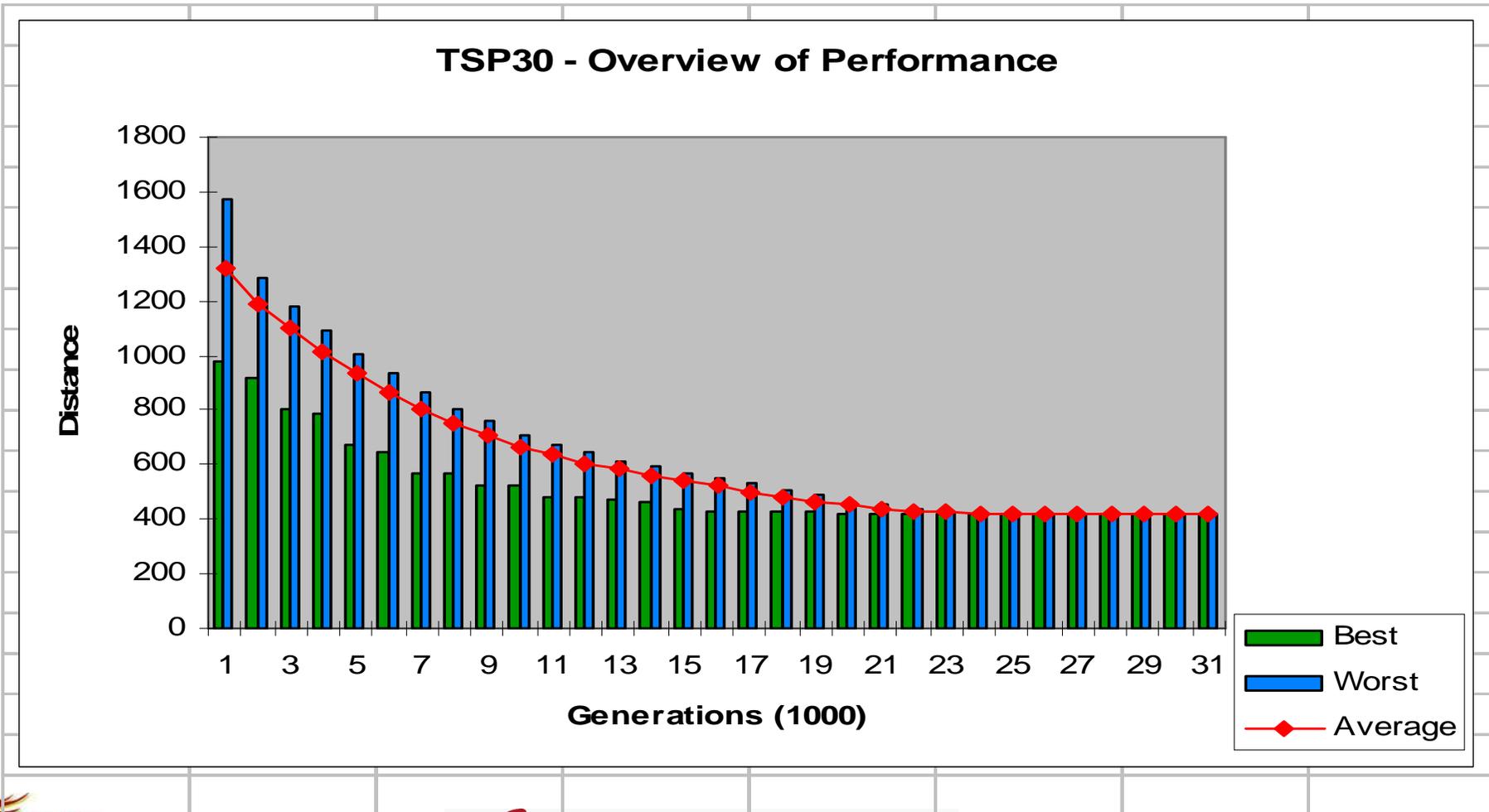
TSP: Terceira solução



TSP: Melhor solução



TSP: Progresso da Evolução



Alguns Tipos de Aplicações

Domain	Application Types
Control	gas pipeline, pole balancing, missile evasion, pursuit
Design	semiconductor layout, aircraft design, keyboard configuration, communication networks
Scheduling	manufacturing, facility scheduling, resource allocation
Robotics	trajectory planning
Machine Learning	designing neural networks, improving classification algorithms, classifier systems
Signal Processing	filter design
Game Playing	poker, checkers, prisoner's dilemma
Combinatorial Optimization	set covering, travelling salesman, routing, bin packing, graph colouring and partitioning

Softwares de Algoritmos Genéticos

- **GENOCOP III**–Genetic Algorithm for Constrained Problems in C (by ZbigniewMichalewicz)
- **DE**–Differential Evolution Genetic Algorithm in C and Matlab(by Rainer Storn). DE has won the third place at the 1st International Contest on Evolutionary Computationon a real-valued function testsuite
- **PGAPack**–Parallel Genetic Algorithm in Fortran and C (from Argonne National Laboratory)
- **PIKAIA**–Genetic algorithm in Fortran 77/90 (by Charbonneau, Knapp and Miller)
- **GAGA**–Genetic Algorithm for General Application in C (by Ian Poole)
- **GAS**–Genetic Algorithm in C++ (by Jelasityand Dombi)
- **GAlib**–C++ Genetic Algorithm Library(by Matthew Wall)
- **Genetic Algorithm in Matlab**(by Michael B. Gordy)
- **GADS**–Genetic Algorithm and Direct Search Toolbox in Matlab(from MathWorks)
- **GEATbx**–Genetic and Evolutionary Algorithm Toolbox for Matlab (by HartmutPohlheim)
- **GAOT**–Genetic Algorithms Optimization Toolboxin Matlab(by Jeffrey Joines)



Alguns Sites Interessantes

- <http://www-2.cs.cmu.edu/Groups/AI/html/faqs/ai/genetic/top.html>
- <http://cs.gmu.edu/research/gag/>
- <http://www-illigal.ge.uiuc.edu/index.php3>
- <http://www.arch.columbia.edu/DDL/cad/A4513/S2001/r7/>
- <http://www.aic.nrl.navy.mil/galist/>
- <http://www.aaai.org/AITopics/html/genalg.html>
- <http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic/ga/0.html>
- <http://psychology.about.com/od/companies/>
- <http://www.nutechsolutions.com/>
- <http://www.autonomoussolutions.com/>
- <http://www.palisade.com/>
- <http://www.optisyn.com/>

